CLUBE

Z80

/83

N.º 10

COORDENADORES: MARIA IRENE • ALBERTO FERNANDES • AV. BOAVISTA, 832 • TELEF. 65127 • 4100 PORTO

CLUBE Z-80

NUMERO 10 / 31 JULHO 1983
AV. BOAVISTA, 832 - 2 T.   Telef.65127
4100 PORTO

N E S T E   N U M E R O

****
P R I M E I R A   P A G I N A
****

INQUERITO

Se ainda nao respondeu ao inquerito enviado com o numero anterior, procure faze-lo com brevidade!

Creia que a sua resposta e muito importante para todos nos e se obtivermos poucas respostas ficaremos com uma certa frustracao relativamente ao trabalho desenvolvido.

PREENCHA E ENVIE-NOS O SEU INQUERITO!

PROGRAMAS PUBLICADOS

O nosso colaborador e associado ROCHA BARBOSA sugeriu que o Clube fornecesse a um preco modico os programas que se publicam nos boletins, GRAVADOS EM CASSETE.

Na opiniao desse associado, os programas extensos podem nao convidar o leitor "mais ocupado" a carregar, linha a linha, o programa no seu computador. Assim, o Clube vai colocar a vossa disposicao cassetes com os programas mais extensos, devidamente gravados e testados. Por exemplo, o leitor que quiser adquirir o programa JOGO DO BANCO que vem publicado neste numero (para o ZX81) bastara remeter Esc. 330$00 e recebera na volta do correio (portes por nossa conta) uma cassete com esse programa.

CURSO DE LINGUAGEM MAQUINA

Tem vindo a ser publicado, desde o numero 7, o Curso de Introducao a Linguagem Maquina da autoria do nosso colaborador e associado FERNANDO PRECES.

Pessoalmente temos seguido com muito interesse a publicacao deste trabalho, mas o Clube e principalmente o autor do Curso precisam conhecer os comentarios daqueles a quem interessa o conhecimento da linguagem maquina.

Escreve FERNANDO PRECES na sua ultima carta: "(...) como o texto esta a ser escrito por grupos de paginas - se bem que no essencial ja esteja programado - pode ser sempre alterado no sentido de oferecer aos leitores aquilo que eles mais procurem."

Se o leitor esta a seguir este texto que publicamos mensalmente, sera importante que COLOQUE AS SUAS QUESTOES e nos TRANSMITA A SUA APRECIACAO GLOBAL E PONTUAL.

==

E N C I C L O P E D I A   D A   L I N G U A G E M
B A S I C

==

JULHO/83 - 1

Iniciaremos neste numero do Jornal do CLUBE Z-80 uma coluna em que tentaremos dar uma resposta a inumeros pedidos e interrogacoes sobre a equivalencia de Instrucoes - Comandos - Palavras Chave que existem na linguagem BASIC, e como funcionam em diversas maquinas. Se os objectivos estao definidos e sao perfeitamente visiveis, o mesmo nao diremos no que respeita a atingi-los com facilidade e se todas as duvidas serao respondidas com nitidez!

BASIC - Trata-se de uma linguagem que pode ser usada para programar um computador.

Foi desenvolvida originalmente no Colegio Dartmouth (E.U.A.) por John Kemeny e Thomas Kurtz, em meados da decada de 60. Os seus criadores pretendiam dar uma ajuda aos estudantes que sentiam dificuldade em assimilar rapidamente COBOL e FORTRAN (que eram (?) as grandes linguagens de programacao da epoca).

Rapidamente, os fabricantes de pequenas maquinas - e nao so - lancaram varias versoes desta linguagem, com os seus microcomputadores, dando assim inicio a pequena Babel que e o BASIC.

A partir dos anos 70, o BASIC foi definitivamente adaptado como linguagem dos "micros" e neste momento nao existe nenhum computador que nao possua um "interpretador" de BASIC.

Em 1978 o American National Standards Institute publicou um padrao desta linguagem para tentar disciplinar a multiplicidade de dialectos aparecidos; quanto a nos, foi sem exito pois existem inumeras versoes de fabricantes de grande prestigio que possuem caracteristicas bastante apreciadas e que nao estao incluidas no padrao ANSI (e esta a denominacao que os curiosos querem designar quando se referem ao BASIC "autentico").

Algumas vantagens do BASIC:

1 - BASIC e "amigavel", ou seja, orientado para as pessoas.

E facil de aprender e, de certo modo, divertida a sua aprendizagem. Qualquer individuo metodico pode, por si proprio, aprender a programacao em BASIC.

2 - A linguagem e flexivel, permitindo desenvolver novos programas e alterar outros existentes com pequenos esforcos.

3 - E uma linguagem universal, disponivel em pequenos e grandes computadores.

4 - A linguagem e largamente independente da maquina. Consequentemente, em geral os programas sao facilmente adaptaveis de maquina para maquina.

A principal desvantagem do BASIC:

E uma linguagem muito indisciplinada!

AS PALAVRAS

Existem quatro categorias de palavras em BASIC:

COMANDOS - Sao palavras interpretadas pelo computador como ordens de execucao. Por exemplo RUN, LIST, etc.

Alguns computadores permitem que os COMANDOS sejam embebidos no programa e nesse caso sao usados como instrucoes.

INSTRUCOES - Trata-se de palavras que aparecem no interior do programa e que compoem todo um percurso detalhado, ao longo do qual o computador toma decisoes e executa tarefas. Por exemplo PRINT A, B, C.

FUNCOES - Palavras que chamam zonas de programa (pre-programadas), regra geral em memoria ROM. Elas executam tarefas relativamente complicadas, tais como determinar um valor trigonometrico, uma raiz quadrada, etc. Exemplo: PRINT LOG(A).

OPERADORES - Sao caracteres que possuem capacidades comparativas ou de modificacao da execucao do programa. Exemplo: virgula, ponto, sinal de igual, etc.

Terminada esta simplista apresentacao de alguns pontos do BASIC, vamos apresentar a primeira palavra-chave:

ABS

Funcao que pertence ao padrao ANSI.

Alguns computadores aceitam A. como alternativa para ABS.

Esta funcao determina o valor ABSoluto de um numero ou de uma variavel numerica. O valor absoluto de um numero e o seu valor sem sinal algebrico + ou -.

Exemplo : PRINT ABS(-20)

Resultado: 20

ABS e capaz de manejar qualquer numero (grande ou pequeno) dentro das capacidades e limitacoes do interpretador do seu computador.

Programa TESTE #1

```
10 REM "ABS-PROGRAMA TESTE"
20 LET X=85
30 PRINT "ABS PASSA NO TESTE
SE",
35 PRINT "TODOS OS VALORES
SAO POSITIVOS"
40 PRINT ABS (-35.2),
50 PRINT ABS (-.032),
60 PRINT ABS (-X)
```

Se o seu computador NAO POSSUI ESTA FUNCAO

Ela pode ser facilmente simulada pela seguinte rotina:

```
10 REM "ABS-SIMULADA"
20 PRINT "ENTRADA DE UM
NUMERO NEGATIVO"
30 INPUT X
40 GO SUB 3000
50 PRINT "O VALOR ABSOLUTO
DE ....."; X; ".E."; Y
60 GO TO 20

3000 REM *ABS (X)*ENTRADA
X*SAIDA Y
3010 LET Y=X
3020 IF X>=0 THEN GO TO 3040
3030 LET Y=-X
3040 RETURN
```

(Continua no prox. numero)

INTRODUCAO A LINGUAGEM MAQUINA
(continuacao)

FERNANDO D'ALMEIDA PRECES

Vamos agora entrar no ja anunciado programa 3, capaz de fazer uma transferencia de bytes em bloco, utilizando uma instrucao LDIR.

A "LDIR" faz parte de um grupo de instrucoes muito importantes no Z80, pois permitem num programa mover blocos de memoria ou ainda pesquisar esses mesmos blocos.

Numa ordem para o Z80 usar a instrucao de movimento de Bloco, o endereco base do Bloco e colocado no registo (HL), o endereco do destino no registo (DE), e a dimensao do Bloco no registo (BC).

Numa ordem para o Z80 usar a instrucao de pesquisar um bloco da memoria, para encontrar um endereco com um determinado conteudo, o endereco base do bloco e colocado no registo (HL), as dimensoes do bloco no registo (BC) e o valor procurado no registo A.

Este grupo de instrucoes divide-se ainda em instrucoes automaticas e nao automaticas.

As automaticas sao aquelas que uma unica instrucao move ou pesquisa um bloco.

As nao automaticas somente movem ou pesquisam um byte, quando da execucao da instrucao. Para ser movido ou pesquisado um certo numero de bytes estas instrucoes precisam de um CICLO.

As instrucoes automaticas sao:

LDIR, LDDR, LPIR e CPDR

As nao automaticas sao:

LDI, LDD, CPI e CPD

Facamos agora a descricao teorica da instrucao que vamos utilizar.

A LDIR e uma instrucao que comeca por mover um byte de (HL) para (DE). Os registos HL e DE sao depois incrementados (somado 1 a cada um deles), o contador de bytes, o registo (BC) decrementado (subtraccao de 1); e o processo desenvolve-se ate BC atingir o valor zero. Esta instrucao pode mover qualquer numero de bytes desde (1 a 65536).

As restantes instrucoes serao detalhadas a medida que forem surgindo no texto.

A nossa primeira experiencia no ambito do programa 3 vai ser de formar um ecran atraves duma rotina de desenhar, memorizar este mesmo ecran num determinado espaco da memoria, e chama-lo em seguida de novo para o ecran, utilizando uma instrucao LDIR.

Atraves da ROTINA ARTE (desenhar o ecran) o leitor pode, utilizando as teclas 1 a 8, imprimir no ecran um a um, qualquer caracter do teclado no sitio exacto aonde o quer colocar. A subrotina 9450 da-lhe as instrucoes necessarias para o fazer, tendo em consideracao que vamos chamar NORTE ao topo superior do ecran. O desenho a memorizar vai ficar pois ao criterio e ao gosto de cada utilizador.

Vamos agora analisar as duas pequenas rotinas em codigo maquina que fazem o trabalho de memorizar ou pintar o ecran.

A primeira, ECRAN/MEMORIA, situa-se na REM 1 (16514).

A segunda, MEMORIA/ECRAN, situa-se na REM 2 (16540).

```
REM 1

 42 - LD HL,(NN) ]  LET HL = PEEK 16396 +
 12 -]           ]    256 * PEEK 16397
 64 -]   16396   ]
 35 - INC HL     ]
 17 - LD DE,NN   ]
 12 -]           ]  LET DE = 31500
123 -]   31500   ]    (ZONA LIVRE DA MEMORIA)
  1 - LD BC,NN   ]
191 -]           ]  LET BC = 703
  2 -]   703     ]
237 -]           ]   TRANSFERENCIA EM BLOCO DO
176 -]   LDIR    ]   ECRAN → MEMORIA
201 - RET            RETURN AO BASIC
```

ROTINA EM BASIC EQUIVALENTE A ESTE C.M.

```
REM ECRAN/MEMORIA
LET L = PEEK 16396
LET H = PEEK 16397
LET HL = 256 * H+L
LET DE = 31500
FOR B(C) = 703 TO 0 STEP - 1
LET A = PEEK HL
POKE DE,A
LET HL = HL+1
LET DE = DE+1
NEXT B
```

DIFERENCA DE TEMPO ENTRE O TRABALHO DESTAS DUAS ROTINAS:
- EM BASIC CERCA DE 2 MINUTOS
- EM C.M. (codigo maquina) INSTANTANEO

```
REM 2

237]                ]  LET DE = PEEK 16396 +
 91]   LD DE,(NN)   ]    256 * PEEK 16397
 12]                ]
 64]   16396        ]
 19    INC DE       ]
 33    LD HL,NN     ]  LET HL = 31500
 12]                ]
123]   31500        ]
  1    LD BC,NN     ]
191]                ]  LET BC = 703
  2]   703          ]
237]                ]  TRANSFERENCIA EM BLOCO
176]   LDIR         ]  DA MEMORIA/ECRAN
201    RET             RETURN AO BASIC
```

ROTINA EM BASIC EQUIVALENTE A ESTE C.M.

```
REM MEMORIA/ECRAN

LET E = PEEK 16396
LET D = PEEK 16397
LET DE = 256 * D+E
LET HL = 31500
FOR B(C) = 703 TO 0 STEP-1
LET A = PEEK HL
POKE DE,A
LET DE = DE+1
LET HL = HL+1
NEXT B
```

A DIFERENCA DO TEMPO DE EXECUCAO E O MESMO ENTRE AS DUAS ROTINAS: APROXIMADAMENTE 2 MINUTOS.

....

TRANSFERENCIA DE BYTES EM BLOCO

PROGRAMA 3

A numeracao das linhas BASIC foi cuidada para que este programa pudesse ser encaixado no nosso programa monitor.

Escreva:

```
300 REM ECRAN/MEMORIA
310 RAND USR 16514
320 STOP
400 REM MEMORIA/ECRAN
410 RAND USR 16540
420 STOP
```

"ROTINA ARTE NO ECRAN"

```
9400 REM ARTE
9401 GOSUB 9450
9402 LET A = 11
9404 LET B = 16
9406 LET A$"00121212"
9407 LET B$"12001221"
9408 PAUSA 300
9409 CLS
9410 INPUT V$
9412 IF INKEY$ = " " THEN PRINT AT A,B;"▫";AT A,B;V$
9414 IF INKEY$ = " " THEN GOTO 9412
9416 IF INKEY$ = "C" THEN GOTO 300
9418 IF INKEY$ = "9" THEN GOTO 9410
9420 LET X = 0
```

```
9422 LET C$ = INKEY$
9424 IF CODE C$>=29 AND CODE C$<=36 THEN LET X = VAL C$
9426 IF X<1 OR X>8 THEN GOTO 9412
9428 LET D = 0
9430 IF A$(X) = "1" THEN LET D = 1
9432 IF A$(X) = "2" THEN LET D = -1
9434 LET A = A+D
9435 LET D = 0
9436 IF B$(X)="1" THEN LET D=1
9438 IF B$(X)="2" THEN LET D=-1
9440 LET B = B+D
9442 IF ABS A>20 THEN LET A = 20
9444 IF ABS B>30 THEN LET B = 30
9446 PRINT AT A,B;V$
9448 GOTO 9412
9450 PRINT "INSTRUCOES"
9452 PRINT,,"1 - DESLOCACAO ESTE"
9454 PRINT,,"2 -      "       OESTE"
9456 PRINT,,"3 -      "       SUL"
9458 PRINT,,"4 -      "       NORTE"
9460 PRINT,,"5 -      "       SUESTE"
9462 PRINT,,"6 -      "       NOROESTE"
9464 PRINT,,"7 -      "       SUDOESTE"
9466 PRINT,,"8 -      "       NORDESTE"
9468 PRINT,,"PRIMA N/L PARA MAIS INSTRUCOES"
9470 INPUT L$
9472 CLS
9474 PRINT,,"PRIMA (C) PARA MEMORIZAR O ECRAN"
9476 PRINT,,"  "   (9) PARA MUDAR O CARACTER"
9478 PRINT,,"  "   N/L PARA COMECAR"
9480 INPUT L$
9482 RETURN

1 REM 12345678901234567890 (20 caracteres)
2 REM 12345678901234567890 (20 caracteres)
```

Chame a rotina 3 (introducao do codigo maquina) e escreva:

16514

42, 12, 64, 35, 17, 12, 123, 1, 191, 2, 237, 176, 201, 256.

Volte a chamar a rotina 3 e escreva:

16540

237, 91, 12, 64, 19, 33, 12, 123, 1, 191, 2, 237, 176, 201, 256.

Grave o programa e siga as instrucoes ja apontadas.
Para transferir o desenho da memoria para o ecran faca GOTO 400.

Este texto de INICIACAO A LINGUAGEM MAQUINA nao tem pretensoes, nesta primeira fase, de apresentar programacao espectacular, mas sim fornecer os rudimentos do funcionamento da maquina, capaz de espevitar a imaginacao de cada utilizador no sentido de ele criar os seus proprios programas com bases teoricas mais solidas. No entanto, ao fim de cada capitulo encontrara o leitor alguns programas de jogos nos quais se fara a aplicacao pratica das instrucoes estudadas.

(CONTINUA NO PROX.NUMERO)

****

S E C C A O   D O   L E I T O R

****

DUVIDAS...SUGESTOES...COMENTARIOS...OPINIOES...DUVIDAS...SUGESTOES

*"Pretendo um programa para saber o custo de um determinado produto e o seu lucro. O caso e o seguinte:

1 Kg. de cafe custa 700$00; a medida para se tirar uma bica ronda as 7 gr.; inclui ainda despesas de agua, gas e vencimento medio dos trabalhadores (19 400$00)."

FERNANDO SOUSA /
/Caldas da Rainha

– Pode resolver este problema do seguinte modo:

```
PRINT "NUMERO MEDIO DE CAFES (DIARIO)"
INPUT X
PRINT "PRECO POR KG DE CAFE"
INPUT P
PRINT "SALARIO MEDIO"
INPUT S
PRINT "VALOR MENSAL GAS"
INPUT G
PRINT "VALOR MENSAL AGUA"
INPUT A
LET CM=30*X
LET PC=CM*0.007
LET VC=P*PC
CLS
PRINT "CUSTO DOS CAFES (MENSAL)=";
LET CF=VC+S+G+A
PRINT CF
PRINT "LUCRO MENSAL MEDIO=";
LET LC=(X*12.5)-CF
PRINT LC
```

– Quanto ao seu programa TORPEDO, depois de o termos passado nao detectamos qualquer tipo de erro. Propomos que verifique o programa de novo, tomando maior atencao com as variaveis.

*"Qual o modo de jogar o OTHELO publicado no numero 8?"

EDUARDO FONTES /
/ Porto

– Solicitamos ao autor do programa (o nosso colaborador Fernando Preces) que, se lhe for possivel, envie instrucoes sobre o programa.

Contudo, no proximo numero publicaremos o REVERSI, identico ao OTHELO.

NOVOS JOGOS PARA O ZX81
=======================

-Simulador de voo...450$00
-Gamao e Dados......450$00

NOVOS LIVROS
============

– PILOTEZ VOTRE ZX81, Patrick Gueulle...250$00

– THE COMPLETE SPECTRUM ROM DISASSEMBLY, Ian Logan e Frank O'Hara...480$00

– THE HOBBIT, J.R.R.Tolkein...590$00

– SPECTRUM HARDWARE MANUAL, Adrian C.Dickens...220$00

– THE SPECTRUM PROGRAMMER, S.M.Gee...300$00

(NOTA:Precos dos livros em fotocopia)

ZX 81

PROGRAMA ..... JOGO do BANCO ...... ZX 81

Adaptado a partir do disco emitido por MICRO-SYSTEMES (Maio 83)

Colaboração de Jorge Costa e Joaquim Magalhães/Porto

INSTRUÇÕES : O jogo consiste em colocar num saco representado pelo simbolo $,
o dinheiro de 1 a 5 dollars (1 , 2 , 3 , 4 , 5) e em seguida depositá-lo
no BANCO, podendo fazer isso de uma vez só ou em várias etapas.
ATENÇÃO : logo que consiga o primeiro dinheiro, aparecerá um grupo de assaltantes aos quais deve escapar e que são representados pelo simbolo "■"
No caso de ser apanhado, o conteudo do saco desaparecerá.
COMANDOS : mudança de direcção : 5 - para a esquerda ; 8 - para a direita
7 - para cima , 6 para baixo, M - para parar.
O objectivo do jogo, será o de depositar o máximo no BANCO.
O jogo apresenta dois tipos de Labirinto que diferem na velocidade, na estrutura do labirinto, na quantidade de bancos e no score máximo, sendo de 30 $ no primeiro e de $ 60 no segundo. Para poder jogar no segundo labirinto terá de obter os $ 3Ø ou se não o conseguir : " BREAK" e GOTO 3ØØØ . (GOTO 2ØØØ para o 1º jogo)

UM CONSELHO : logo que o saco contenha dinheiro, a prisão será aberta, aumentando a dificuldade do jogo, mas se fôr assaltado e o saco estiver vazio, o assaltante voltará de novo para a prisão.

```
  3 SAVE "BANCO"
 20 SLOW
 30 PRINT AT 0,10;"banco"
 40 PRINT AT 4,0;"COM O SACO ""$"" DEVE APANHAR A   MAIOR QUANTIDADE DE DINHEIR
O    POSSIVEL, MOVIMENTANDO-SE NO     INTERIOR DE UM LABIRINTO.          EM SEGUIDA
DEVE COLOCA-LO NO    BANCO, SEM SE DEIXAR APANHAR     PELOS ASSALTANTES """".
   CASO CONTRARIO O SEU DINHEIRO   DESAPARECERA"
110 LET C$="CD8A75473E0DB8200801070036003333C978CB7B2007CB7F28012FA3C9CB7F18FB"

120 LET R=30000
130 GOSUB 5000
210 LET C$="CD8A75CB7B200BCB7F28042FB32FC9B300C9CB7F20022FA3C9"
220 LET R=30035
230 GOSUB 5000
310 LET C$="CD8A75CB7B200BCB7F20042FB32FC9B300C9CB7F28022FA3C9"
320 LET R=30060
330 GOSUB 5000
410 LET C$="CB3F300DCB3930051E870018101E02180CCB3930041E0418041E01000006002600
9CB25CB25CB25292909C6014F09ED4B0C40097EC9"
420 LET R=30090
430 GOSUB 5000
440 PRINT AT 14,0;"PARA CONTROLO DO SACO           USE:"
450 PRINT AT 16,8;"5 : ▘ ESQUERDA"
460 PRINT AT 17,8;"6 : ▘ P/BAIXO"
470 PRINT AT 18,8;"7 : ▘ P/CIMA"
480 PRINT AT 19,8;"8 : ▘ DIREITA"
500 PRINT AT 20,8;"M : ▘ PARAR"
510 PRINT AT 21,8;"SPACE : PARAR O JOGO"
610 LET C$="3E00321F6E"
620 LET R=30240
630 GOSUB 5000
640 LET C$="16003A226E4F3A216ECD3075280A16013A246EED4432246E3A236E4F3A206ECD307
52816CB4228083A246EED4432246E3A256EED4432256E1836CB4220323A236E4F3A216ECD307520D
E06FF000000000010F9007ECD5675773A226E4F3A206ECD6C75773A216E32206E3A236E32226E3A2
06E21246E8632216E3A226E238632236E"
650 GOSUB 5000
660 CLS
```

OBSERVAÇÃO Linha 4Ø .... ""$"" ....
.............................."" ""

linha 45Ø a 5ØØ o simbolo grafico é obtido com Graphics e $ (shift)

```
 670 PRINT AT 1,0;"O JOGO TEM DUAS APRESENTACOES";AT 5,1;"A PRIMEIRA FACIL:";AT
7,10;"1' 30 &";AT 9,1;"A SEGUNDA PARA BONS JOGADORES:";AT 11,10;"2' 60 &"
 680 PRINT AT 20,0;"ATENCAO. ESPERE UM MOMENTO"
 710 LET M$="28272A2A29262A2A2B2B292728262726292826 2A272829"
 720 GOSUB 5050
 730 CLS
 740 FAST
 810 LET M$="2E2D30302F2C303031312F2D2E2C2D2C2F2E2C302D2E2F"
 820 GOSUB 5050
 910 LET M$="343336363532363637373533343233323534323633 3435"
 920 GOSUB 5050
1010 LET C$="3A1F6ECB4FC23C783C321F6EC325763E00321F6E673A1E6E6FCB7F280226FF01000
03A2540573E7FBAC83EF7BA200521FFFF18203A2640573EF7BA20052101001812 3EEFBA200521DFF
F18083EDFBA2003212100"
1020 GOSUB 5000
1030 LET C$="457DB7CA2076ED5B1C6E197E4F3E00BD20040E06183379B720040E00182B3E9DB92
0040E0118223E9EB920040E0218193E9FB920040E0318103EA0B920040E0418073EA1B920160E05"
10[illegible] GOSUB 5000
1050 LET C$="368D221C6E78321E6EEB3600060079B7CA2076C93A1F6EB720113C321F6EB8CA7E7
83EFFB8CA7E78C35C783A25404F3E7FB92004010000C93EFFB8CA6A783E21B8CA74783E01B8CA7E7
8C35C78"
1060 GOSUB 5000
2005 CLEAR
2010 FAST
2020 LET H$="4300633905073956583508113552553113153148502716192744472321232340421
93939182939070024072958020063"
2030 GOSUB 5100
2040 LET V$="0002430510390807351310311607272110232402192907184207234710275007315
51035580739630243"
2050 GOSUB 5200
2055 LET V=155
2060 LET X=INT (RND*23+1)
2065 LET Y=INT (RND*4+37)
2070 GOSUB 5500
2080 SLOW
2100 LET LV=11
21[illegible] LET CV=12
2120 LET LD=14
2130 LET CD=19
2140 LET DX=255
2150 GOSUB 5800
2200 PRINT AT 13,15;"::::";AT 14,15;"::::  :";AT 15,15;"BANCO "
2210 PRINT AT 1,3;"5";AT 1,28;"5";AT 3,5;"4";AT 3,26;"4";AT 5,7;"3";AT 5,24;"3";
AT 7,9;"2";AT 7,22;"2";AT 9,11;"1";AT 9,20;"1";AT 11,12;"&"
2220 PRINT AT 14,19;" ";AT 16,15;":::::::";AT 17,15;":::::::"
2500 LET U=USR 30240
2510 IF U=0 THEN GOTO 2700
2520 IF U=6 THEN GOTO 2800
2530 IF U=7 THEN GOTO 2900
2550 UNPLOT 24,6
2560 UNPLOT 24,5
2570 LET VAL=VAL+U
2580 LET C$(1 TO 2)=STR$ (VAL)
2590 FOR I=P+3 TO VAL+2
2600 LET C$(I)="$"
2610 NEXT I
2620 PRINT AT 21,0;C$
2630 LET P=VAL
2640 LET V=V+1
```

```
2650 GOTO 2500
2710 POKE 28190,0
2720 GOTO 2500
2805 PRINT AT 16,15;"DEPOS$"
2810 LET DEP=DEP+VAL
2820 LET C$=""
2830 LET VAL=0
2840 LET P=0
2850 PRINT AT 17,17;DEP;"$";AT 21,0;C$;AT 16,15;"::::::"
2860 PLOT 24,6
2865 PLOT 24,5
2870 IF DEP=30 THEN GOTO 3000
2880 IF V=10 THEN GOTO 2100
2890 GOTO 2500
2910 PLOT 24,5
2915 PLOT 24,6
2920 LET A=A+6
2930 IF A>28210 THEN LET A=28192
2940 UNPLOT PEEK A,43-PEEK (A+2)
2945 LET X=INT (RND*23+1)
2950 LET Y=INT (RND*4+37)
2955 GOSUB 5600
2960 LET C$=""
2970 LET VAL=0
2975 LET P=0
2980 PRINT AT 21,0;C$
2985 IF V=10 THEN GOTO 2100
2990 GOTO 2500
3005 CLS
3010 PRINT AT 5,13;"BRAVO";AT 10,0;"AGORA PODE FAZER MELHOR";AT 20,0;"ATENCAO AO
 INICIO DA SEGUNDA    APRESENTACAO DO JOGO"
3015 PAUSE 200
3020 FAST
3025 CLS
3030 CLEAR
3040 LET H$="4300633612343405092839552706172662632539522331322127281923241609521
56263094952095962060645020038025163"
3050 GOSUB 5100
3060 LET V$="2341420003420506330811160935391307111611151827332107112227332411152
62733290711302733321115342535370711390203391724392839401115433239450711473439480
615500203521724513639530609553839580609591228630314632742"
3070 GOSUB 5200
3075 LET X=INT (RND*9+40)
3080 LET Y=INT (RND*8+19)
3085 LET V=200
3090 GOSUB 5500
3100 SLOW
3110 LET LV=8
3120 LET CV=18
3130 LET LD=4
3140 LET CD=5
3150 LET DX=1
3160 GOSUB 5800
3200 PRINT AT 5,3;"BANCO ";AT 6,3;"::::::";AT 7,3;AT 7,3;"::::::";AT 20,20;"DEPO
S";AT 9,31;"D";AT 10,31;"E";AT 11,31;"P";AT 12,31;"O";AT 13,31;"S"
3210 PRINT AT 15,22;"5";AT 14,25;"5";AT 12,13;"5";AT 6,20;"5";AT 7,16;"4";AT 15,
18;"4";AT 16,25;"4";AT 5,22;"4";AT 4,24;"3";AT 18,30;"3";AT 7,14;"3";AT 15,14;"3
";AT 3,26;"2";AT 7,12;"2";AT 15,10;"2";AT 18,25;"2";AT 3,3;"1";AT 2,28;"1";AT 7,
10;"1";AT 15,6;"1";AT 8,18;"&"
3500 LET U=USR 30240
3510 IF U=0 THEN GOTO 3700
3520 IF U=6 THEN GOTO 3800
3530 IF U=7 THEN GOTO 3900
3550 PRINT AT 10,19;" ";AT 10,26;" ";AT 13,22;" "
```

```
3560 LET VAL=VAL+U
3570 LET C$(1 TO 2)=STR$ (VAL)
3580 IF VAL>30 THEN GOTO 3620
3590 FOR I=P+3 TO VAL+2
3600 LET C$(I)="$"
3610 NEXT I
3620 PRINT AT 21,0;C$
3630 LET P=VAL
3640 LET V=V+1
3650 GOTO 3500
3710 POKE 28190,0
3720 GOTO 3500
3805 PRINT AT 6,3;"DEPOS$"
3810 LET DEP=DEP+VAL
3820 LET C$=""
3830 LET VAL=0
3840 LET P=0
3850 PRINT AT 7,5;DEP;"$";AT 21,0;C$;AT 6,3;"::::::"
3860 PRINT AT 10,19;"";AT 10,26;"";AT 13,22;""
3870 IF DEP=70 THEN GOTO 2000
3880 IF V=20 THEN GOTO 3100
3890 GOTO 3500
3910 PRINT AT 10,19;"";AT 10,26;"";AT 13,22;""
3920 LET A=A+6
3930 IF A>28210 THEN LET A=28192
3940 UNPLOT PEEK A,43-PEEK (A+2)
3945 LET X=INT (RND*9+40)
3950 LET Y=INT (RND*8+19)
3955 GOSUB 5600
3960 LET C$=""
3970 LET VAL=0
3975 LET P=0
3980 PRINT AT 21,0;C$
3985 IF V=20 THEN GOTO 3100
3990 GOTO 3500
5005 FOR X=1 TO LEN C$-1 STEP 2
5010 POKE R,16*(CODE C$(X)-28)+CODE C$(X+1)-28
5020 LET R=R+1
5030 NEXT X
5040 RETURN
5055 LET P$="040812171A1E2A2F323740445B5F66696C6F7275797C81"
5060 FOR X=1 TO LEN P$-1 STEP 2
5070 LET C=2*(16*(CODE P$(X)-28)+CODE P$(X+1)-28)
5080 LET C$(C-1 TO C)=M$(X TO X+1)
5090 NEXT X
5095 GOTO 5000
5105 FOR H=1 TO LEN H$-1 STEP 6
5110 FOR X=VAL H$(H+2 TO H+3) TO VAL H$(H+4 TO H+5)
5120 PLOT X,VAL H$(H TO H+1)
5130 NEXT X
5140 NEXT H
5150 RETURN
5205 FOR V=1 TO LEN V$-1 STEP 6
5210 FOR Y=VAL V$(V+2 TO V+3) TO VAL V$(V+4 TO V+5)
5220 PLOT VAL V$(V TO V+1),Y
5230 NEXT Y
5240 NEXT V
5250 RETURN
```

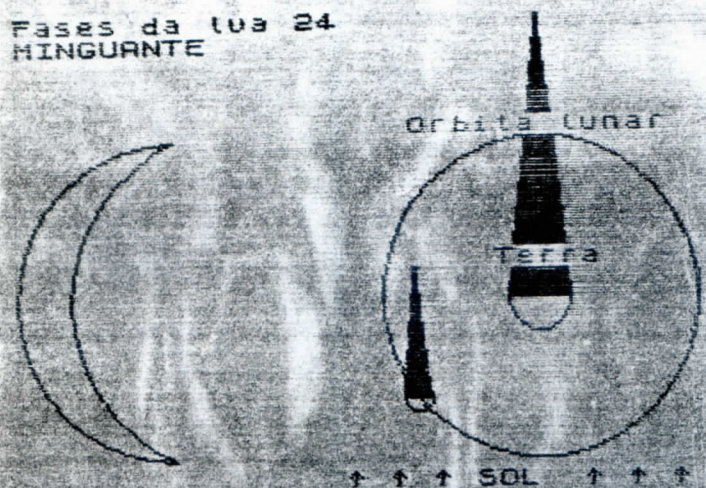
```
5510 FOR A=28192 TO 28210 STEP 6
5520 GOSUB 5600
5530 NEXT A
5540 POKE 30320,V
5550 POKE 30450,V
5560 POKE 30580,V
5570 POKE 30710,V
5580 RETURN
5600 LET DX=1
5610 LET DY=1
5640 IF INT (RND*1) THEN LET DX=-1
5650 IF INT (RND*1) THEN LET DY=-1
5660 POKE A,X
5670 POKE A+1,X+DX
5680 POKE A+2,Y
5690 POKE A+3,Y+DY
5700 POKE A+4,DX
5710 POKE A+5,DY
5720 RETURN
5810 LET DFILE=PEEK 16396+256*PEEK 16397
5820 LET X=DFILE+1+33*LV+CV
5830 POKE 28188,X-256*INT (X/256)
5840 POKE 28189,INT (X/256)
5850 POKE 28190,DX
5860 LET X=DFILE+1+33*LD+CD
5870 POKE 30863,X-256*INT (X/256)
5880 LET VAL=0
5890 LET DEP=0
5900 DIM C$(32)
5910 LET P=0
5920 LET V=0
5930 RETURN
```

SPECTRUM 16K

****

F A S E S   D A   L U A

****

In.
POPULAR COMPUTING, Junho/83

```
   4 LET G=192: LET H=69: LET J=
58: LET K=52: LET M=108: LET Z=0
   6 FOR D=0 TO 28
   8 CLS : CIRCLE G,H,J: CIRCLE
G,H,12
  10 LET P=PI: LET A=(D-7)/14*P
  11 LET C=G+J*COS A: LET E=H+J*
SIN A: CIRCLE C,E,5
  13 FOR N=.5 TO 5: PLOT C+N,E:
DRAW 0,K-10*N: PLOT C-N,E: DRAW
0,K-10*N: NEXT N
  15 FOR N=.5 TO 12: PLOT G+N,H:
 DRAW 0,M-9*N: PLOT G-N,H: DRAW
0,M-9*N: NEXT N
  20 PRINT "Fases da lua ";D
  23 PRINT FLASH 1;("LUA NOVA "
AND (D=0 OR D=28))+("1o " AND D=
7)+("2o " AND D=21)+("QUARTO " A
ND (D=7 OR D=21))+("CHEIA " AND
D=14)+("CRESCENTE " AND (D>0 AND
 D<14 AND D<>7))+("MINGUANTE " A
ND (D>14 AND D<28 AND D<>21))+""
  26 PRINT FLASH 1;("Possivel ec
lipse do sol" AND (D=0 OR D=28))
+("Possivel eclipse da lua" AND
D=14)
  28 PRINT AT 11,22;"Terra";AT 5
,18;"Orbita lunar"
  29 PRINT AT 21,13;" ↑ ↑ ↑ SOL
↑ ↑ ↑ "
  36 IF D>14 THEN LET P=-P
  40 PLOT J,10: DRAW 0,J*2,P
  50 LET B=D-7: LET X=2.5
  60 IF B>7 THEN LET B=B-14
  70 LET N=X*ATN (PI/180*-B*25)
  80 PLOT J,10: DRAW 0,J*2,N
  85 IF Z=1 THEN GO TO 200
  90 PAUSE 300: NEXT D: LET Z=1
 200 INPUT "Seleccione o dia ";D
 210 GO TO (D>=0 AND D<=28)*8
 250 SAVE "LUA" LINE 1
```

A lua e o nosso vizinho mais proximo do espaco, cuja influencia recai sobre nos desde a antiguidade.

Este programa para o SPECTRUM 16K explica as diferentes fases da lua no mes e o seu movimento a volta da terra; mostra ainda os eclipses do sol e da lua e a razao do eclipse total do sol em muitos lugares da terra.

Ao executar a instrucao "RUN" podera ver o ciclo completo da lua.
Terminando, o programa convida-o a seleccionar um dia do mes - <0 ou >28 - para verificar a fase em que a lua se encontra.

Pode ainda, com a sua imaginacao, completar o programa com cor e som, tendo o cuidado de nao alterar os seus efeitos graficos.

SPECTRUM 48K

CAMPO DE MINAS

In.
YOUR COMPUTER, Junho 1983

NOTAS:

10 - 70 define graficos

80 - 85 instrucoes (gosub) e entrada do nivel

100 - 175 representacao do quadro do jogo

190 localizacao das minas

192 - 200 representacao do "Tank"

208 - 225 deteccao das minas

250 - 285 movimento e verificacao da posicao das minas

290 - 330 localizacao do soldado

350 - 362 rotina da explosao

400 - 470 "repeticao"

500 - 615 jogo terminado com sucesso

650 - 680 marcha funebre

1000 - 1110 leitura das instrucoes

1500 - 1590 limite da area de jogo

```
  10 FOR n=144 TO 148: FOR f=0 T
O 7
  20 READ a: POKE USR (CHR$ n)+f
,a: NEXT f: NEXT n
  30 DATA 0,0,0,60,126,255,255,0
  40 DATA 126,255,189,189,255,25
5,231,231
  50 DATA 0,28,36,66,68,185,90,6
0
  60 DATA 24,60,126,60,255,255,1
02,102
  70 DATA 156,92,40,31,28,36,66
  80 CLS : PRINT AT 10,1;"QUER L
ER AS INSTRUCOES? (S/N)"
```

```
82 PAUSE 1: PAUSE 0: IF INKEY$="S" OR INKEY$="s" THEN GO SUB 1000
94 CLS : PRINT AT 10,8;"ESCOLHA O NIVEL": INPUT "NIVEL?(1-9) ";d
96 IF d<1 THEN GO TO 83
98 CLS : PRINT AT 10,7;"ESPERE UM MOMENTO"''"ATENCAO A LOCALIZACAO DAS MINAS"
100 FOR n=33940 TO 34510: POKE n,0: NEXT n
115 LET sc=0
120 LET co=7-d: IF co<4 THEN LET co=6
125 BORDER co: PAPER co: INK 7-co: CLS
130 FOR n=0 TO 31: BEEP .005,n: PRINT INK 2;AT 1,n;"@"
135 PRINT INK 2;AT 21,n;"@": NEXT n
140 FOR f=1 TO 21: BEEP .005,f: PRINT INK 2;AT f,0;"@": PRINT INK 2;AT f,31;"@": NEXT f
150 PRINT PAPER co; INK 1;AT 21,14;"  ";AT 1,15;"█"
160 LET a$="                    <Sem minas >               "
170 FOR n=1 TO 30: PRINT PAPER co; INK n/5;AT 20,1;a$(n TO 30);AT 2,n;a$(16 TO 46-n): BEEP .02,11-n/3: BEEP .02,n/3: NEXT n
175 LET po=555: LET co=7-d
190 FOR n=1 TO 20+(d*2): LET a=INT (509*RND)+1: POKE (34000+a),21: NEXT n
192 LET ma=132: LET q=0: LET d=d+1: LET x=21: LET y=15
193 IF d>4 THEN GO SUB 290
195 LET m=0: IF co<3 THEN LET co=6
200 PRINT INK 1; PAPER 6; BRIGHT 1;AT x,y;"*"
205 IF INT (RND*4)=2 THEN LET m=1
208 IF PEEK (34000+po)=21 THEN GO TO 350
210 IF PEEK (33999+po)=21 THEN LET m=m+1
215 IF PEEK (34001+po)=21 THEN LET m=m+1
220 IF PEEK (34030+po)=21 THEN LET m=m+1
225 IF PEEK (33970+po)=21 THEN LET m=m+1
228 PRINT AT x,y; PAPER 6; BRIGHT 1;" "
230 LET k$=INKEY$
231 IF CODE k$<53 OR CODE k$>56 THEN GO TO 238
232 IF k$<>"" THEN LET ma=ma-1
233 IF x=2 AND y=16 AND k$="7" THEN GO TO 600
234 GO SUB (VAL k$*10)+200
235 LET q=q+1
237 POKE 35999+q,VAL k$
238 PRINT PAPER m; INK 7;AT 0,10;m;" DETECTOR DE MINAS": IF m<>1 THEN PRINT PAPER m; INK 7;AT 0,30;"S"
240 IF d>4 THEN GO SUB 310
245 GO TO 195
250 LET y=y-1: IF y<1 OR x=21 AND y<14 THEN GO TO 1500
255 LET po=po-1: RETURN
260 LET x=x+1: IF x>20 THEN GO TO 1500
265 LET po=po+30: RETURN
270 LET x=x-1: IF x<2 THEN GO TO 1500
275 LET po=po-30: RETURN
280 LET y=y+1: IF y>30 OR x=21 AND y>16 THEN GO TO 1500
285 LET po=po+1: RETURN
290 LET bx=INT (RND*17)+3: LET by=INT (RND*30)+1
300 POKE 35000,0: PRINT AT bx,by;"X": RETURN
310 IF PEEK 35000=1 THEN RETURN
315 IF x=bx AND y=by THEN GO TO 330
320 RETURN
330 BEEP .5,20: POKE 35000,1: RETURN
350 LET j=(7.97*y)+4: LET k=175-(7.95*x+1)
355 BEEP 1,-30: BEEP 1,-38
360 FOR n=1 TO 20: PLOT j,k: DRAW INK 2;10*SIN n,10*COS n: NEXT n
361 FOR n=0 TO 6: PRINT INK n;AT 10,14;"█████": BEEP .25,-20-n: NEXT n: PAUSE 1: PAUSE 50
362 GO SUB 650
365 PAUSE 1: PAUSE 150: GO SUB 400
370 PRINT INK 0;AT 10,4;"Outra tentativa? (S/N)": PAUSE 1: PAUSE 0
375 IF INKEY$="N" OR INKEY$="n" THEN STOP
380 RUN 83
400 CLS : FOR N=0 TO 31: PRINT INK 2;AT 1,n;"@": PRINT INK 2;AT 21,n;"@": NEXT n
410 FOR f=1 TO 21: PRINT INK 2;AT f,0;"@": PRINT INK 2;AT f,31;"@": NEXT f
415 LET g=0
420 FOR x=3 TO 19: FOR y=1 TO 30
430 LET g=g+1: IF PEEK (34000+g)=21 THEN PRINT INK 0;AT x,y;"▲"
435 NEXT y: NEXT x
440 PRINT AT 0,10;"█████████";AT 0,0;"Score:";sc;AT 1,15;"█";AT 21,14;"  "
442 IF d>4 THEN PRINT AT bx,by;"X"
445 LET x=21: LET y=15
450 FOR a=0 TO q-1: LET r=PEEK (36000+a)
452 IF r=6 AND x<20 THEN LET x=x+1
453 IF r=7 AND x>2 THEN LET x=x-1
454 IF r=8 AND y<30 THEN LET y=y+1
455 IF r=5 AND y>1 THEN LET y=y-1
457 PRINT INK 1; BRIGHT 1; PAPER 6;AT x,y;"*"
460 PAUSE 10: PRINT AT x,y; PAPER 6; BRIGHT 1;" "
465 NEXT a
470 GO SUB 650: RETURN
500 FOR f=0 TO 10 STEP 8: RESTORE 510: FOR n=0 TO 7: READ a,b
505 BEEP a*2,b+f: NEXT n: NEXT f
510 DATA .1,0,.05,0,.05,0,.05,5
520 DATA .05,0,.05,5,.25,9,.25,5
530 RETURN
600 LET d=d+1: LET sc=sc+ma: GO SUB 500: PRINT AT 0,1;"Score:";sc
610 IF PEEK 35000=1 THEN LET sc=sc+50: PAUSE 1: PAUSE 150: GO TO 120
615 PAUSE 1: PAUSE 150: GO TO 120
650 RESTORE 670: FOR n=1 TO 11: READ a,b
660 BEEP a,b: NEXT n: RETURN
670 DATA .5,-3,.55,-3,.2,-3,.65,-3
680 DATA .5,0,.25,-1,.5,-1,.25,-3,.5,-3,.25,-4,.75,-3
1000 INK 1: BORDER 6: PAPER 6: CLS
1010 PRINT AT 1,3;"███ ███ █ █ ███  ███"
1020 PRINT AT 2,2;"█  █ ██ ███ █  █ █  █"
1030 PRINT AT 3,2;"█  █ ██ █ █ █  █ █  █"
1040 PRINT AT 4,2;"█  ████ █ ███ █  █"
```

```
1050 PRINT AT 5,2;"[graphics]"
1060 PRINT AT 6,2;"[graphics]"
1080 PRINT AT 7,3;"[graphics]"
1090 PRINT AT 9,12;"[graphics]"
1100 PRINT AT 10,12;"[graphics]"
1110 PRINT AT 11,12;"[graphics]"
1120 PRINT AT 12,12;"[graphics]"
1130 PRINT AT 14,2;"[graphics]"
1132 PRINT AT 15,2;"[graphics]"
1133 PRINT AT 16,2;"[graphics]"
1135 PRINT AT 17,2;"[graphics]"
1137 PRINT AT 18,2;"[graphics]"
1138 PRINT AT 19,2;"[graphics]"
1139 PRINT AT 20,2;"[graphics]"
1141 GO SUB 500
1145 CLS : PRINT AT 0,10;"INSTRU
COES"
1150 PRINT AT 2,0;"Este jogo tem
 como objectivo a  conducao de u
m tank ate ao       aquartelament
o mais proximo."
1155 PRINT "A dificuldade encont
ra-se na     passagem por um camp
o minado.    Como unico meio de a
poio, tem    um detector de minas
, Este       informa-lo-a da apro
ximidade     de uma ou duas minas
."
1160 PRINT "Tera bonus de pontua
cao se       conseguir captar o s
oldado que   lhe aparecer defende
r o campo    de minas."
1165 PRINT   "CONTROLOS.": PRINT
 "5- esquerda            6- descer
  7- subir               8- direita"
1167 PRINT AT 21,0; FLASH 1; INV
ERSE 0;"QUALQUER TECLA PARA CONT
INUAR"
1170 PAUSE 1: PAUSE 0: CLS : GO
TO 83
1500 LET j=(7.97*y): LET k=175-(
7.95*x+4)
1510 PLOT j,k
1515 FOR n=0 TO 7: PLOT j,k: BEE
P .001,20: DRAW INK 6;5*COS n-5,
5*SIN n+4: NEXT n
1530 GO SUB 650
1540 GO SUB 400
1550 PRINT AT 10,5;"Outra tentat
iva? (S/N)": PAUSE 1: PAUSE 0
1560 IF INKEY$="n" OR INKEY$="N"
 THEN STOP
1590 RUN 83
```

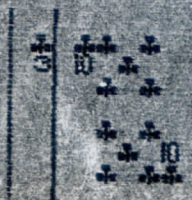
GRAFICOS

- ▲ - Graphics A
- ▲ - Graphics B
- ♣ - Graphics C
- ♠ - Graphics D
- ✱ - Graphics E

===========
SPECTRUM 48K
===========

****
V I N T E   E   U M
****

In. SINCLAIR USER, Jan. 83

(Jogo identico ao tradicional "sete e meio")

```
   1 CLS
   5 GO SUB 2260
  10 BORDER 4: PAPER 7: INK 0
  20 CLS
  30 LET z=0
  40 PAPER 7
  50 PRINT AT 1,10;"VINTE E UM"
  60 PRINT AT 2,10;"▀▀▀▀▀▀▀▀▀▀"
  70 LET x=1
  80 PRINT ''
  90 PRINT "    O VALOR DAS CARTA
S:"
 100 PRINT ''
 110 PRINT "    ""A""= 1 OU 11"
 120 PRINT '"(O SPECTRUM USA O M
ELHOR VALOR)"
 140 PRINT ''
 150 PRINT "    ""10""-""J""-""Q"
"-""K"" VALEM 10"
 160 PRINT ''
 170 PRINT "AS OUTRAS CARTAS = A
O SEU NUMERO"
 190 IF PEEK (USR "a")<>0 THEN G
O SUB 2260
 200 PRINT AT 20,0; FLASH 1;" Qu
alquer tecla para continuar! "
 215 LET y$=INKEY$: IF y$="" THE
N GO TO 210
 220 POKE 23692,42
 240 CLS : PRINT TAB 1;"O object
ivo do jogo:"''" Conseguir com d
uas ou mais       cartas atingir
 valor 21 ou o  mais proximo po
ssivel."
 250 PRINT ''TAB 1;"Antes do in
icio do jogo sera     pedida a qu
antia que deseja     arriscar, n
ao podendo          ultrapassar
 o dinheiro que      possui."''
 Principia com (1000$)"
 270 PRINT AT 15,1;"Em caso de e
mpate ganha o        SPECTRUM"
 300 LET st=1000
 310 DIM a$(53,5)
 320 LET c=0
 340 LET s$="♣♦♥♠"
 350 LET d$="11A022033044055066
077088099100100J100Q10K"
 370 FOR s=1 TO 4
 380 LET a=2
 390 IF s=1 OR s=4 THEN LET a=0
 400 FOR n=1 TO 13
 410 LET c=c+1
 420 LET a$(c)=d$(n*3-2 TO n*3)+
s$(s)+STR$ a
 430 NEXT n: NEXT s
 460 PRINT AT 20,0; FLASH 1;"
qualquer tecla para comecar   "
 470 PAPER 4
 480 IF INKEY$="" THEN GO TO 480
 490 CLS
 500 RANDOMIZE
 530 GO SUB 800
 540 CLS
 550 GO TO 930
 570 LET e$="        "
 590 IF z=1 THEN LET e$="▒▒▒▒▒▒▒▒
"
 600 PAPER 7
 610 FOR i=0 TO 8: PRINT AT v+i,
h;e$: NEXT i
 620 IF z=1 THEN GO TO 770
 630 LET f=CODE a$(c,3): LET f$=
a$(c,4)
 640 INK VAL a$(c,5)
 650 PRINT AT v+1,h+1;f$;AT v+7,
h+5;f$
 660 PRINT AT v+2,h+1;a$(c,3);AT
 v+6,h+5;a$(c,3)
 670 IF f=55 OR f=56 OR f=57 OR
f=151 THEN PRINT AT v+1,h+2;f$;"
 ";f$;AT v+7,h+2;f$;" ";f$
 680 IF f=52 OR f=53 OR f=54 THE
N PRINT AT v+2,h+2;f$;" ";f$;AT
v+6,h+2;f$;" ";f$
 690 IF f=151 OR f=50 OR f=51 TH
EN PRINT AT v+2,h+3;f$;AT v+6,h+
3;f$
 700 IF f=56 OR f=57 OR f=58 OR
f=151 THEN PRINT AT v+3,h+2;f$;"
 ";f$;AT v+5,h+2;f$;" ";f$
 710 IF f=54 THEN PRINT AT v+4,h
+2;f$;" ";f$
 720 IF f=65 OR f=51 OR f=53 OR
f=57 THEN PRINT AT v+4,h+3;f$
 730 IF f=55 THEN PRINT AT v+3,h
+3;f$;AT v+5,h+2;f$;" ";f$
 740 IF f=74 THEN PRINT AT v+3,h
+4;"▌";AT v+4,h+2;"▖ ▌";AT v+5,h
+2;"▝▀▘"
 750 IF f=81 THEN PRINT AT v+3,h
+2;"▞▀▚";AT v+4,h+2;"▌▖▐";AT v+5
,h+2;"▚▄▞";AT v+6,h+4;"▘"
 760 IF f=75 THEN PRINT AT v+3,h
+2;"▌▗▘";AT v+4,h+2;"▛▖";AT v+5,
h+2;"▌ ▚"
 770 PAPER 4: INK 0
 780 RETURN
 790 CLS
 800 PRINT AT 10,10; PAPER 7; FL
ASH 1;"VINTE E UM";AT 12,14;"21"
 810 FOR c=53 TO 2 STEP -1
 820 BEEP .01,c
 830 LET a=(52-(RND*8))+1
 840 LET a$(c)=a$(a): LET a$(a)=
a$(c-1)
 850 NEXT c
 860 FOR c=2 TO 53
 870 BEEP .02,c
 880 LET a$(c-1)=a$(c)
 890 NEXT c
 900 LET c=1
 910 PRINT AT 10,10;"          "
;AT 12,14;"  "
 920 RETURN
 930 IF c>40 THEN GO SUB 790
 940 LET z=1: LET d=0: LET da=0:
 LET p=0: LET pa=0
 950 PAPER 7
 970 LET t$=a$(c)
 980 LET v=1: LET h=1
 990 GO SUB 570
1000 LET z=0
1010 LET d=d+VAL a$(c,1 TO 2)
1020 IF VAL a$(c,1 TO 2)=11 THEN
 LET da=da+1
1030 LET c=c+1
1040 LET v=13
1050 GO SUB 570
1060 LET p=p+VAL a$(c,1 TO 2)
1070 IF VAL a$(c,1 TO 2)=11 THEN
 LET pa=pa+1
1080 LET c=c+1
1090 PRINT AT 14,15;"Tem ";st;"$
"
1100 PRINT AT 16,15;"A sua apost
a ?"
1110 INPUT b$: IF b$="" THEN GO
TO 1110
1120 PRINT AT 14,8;"
             ";AT 16,8;"
             "
1130 FOR i=1 TO LEN b$
1140 IF CODE b$(i)=46 THEN GO TO
 1230
1150 IF CODE b$(i)<48 OR CODE b$
(i)>57 THEN GO TO 1200
1160 NEXT i
1170 LET bet=VAL b$
1180 IF bet<=st THEN GO TO 1250
1190 PRINT AT 14,9;"Nao pode apo
star mais";AT 16,9;"que o dinhei
ro que tem!"
1200 BEEP .8,-10: BEEP .3,-30
1210 PRINT AT 14,8;"
             ";AT 16,8;"
             "
1220 GO TO 1090
1240 GO TO 1200
1250 PRINT AT 11,0;"apostou ";be
t;"$"
1260 LET v=1: LET h=3
1270 GO SUB 570
1280 LET d=d+VAL a$(c,1 TO 2)
1290 IF VAL a$(c,1 TO 2)=11 THEN
 LET da=da+1
1300 IF da=2 THEN LET d=12: LET
da=1
1310 LET c=c+1
1320 LET v=13
1330 GO SUB 570
1340 LET p=p+VAL a$(c,1 TO 2)
1350 IF VAL a$(c,1 TO 2)=11 THEN
```

```
LET pa=pa+1
1360 IF pa=2 THEN LET p=12: LET
pa=1
1370 LET c=c+1
1380 LET pc=2: LET dc=2
1390 LET h=h+2
1400 IF p=21 AND pc=2 THEN GO TO
 1610
1410 PRINT AT 11,15; PAPER 7; FL
ASH 1;"outra carta?(s/n)"
1420 LET c$=INKEY$
1430 IF NOT (c$="s" OR c$="S" OR
 c$="n" OR c$="N") THEN GO TO 14
20
1440 IF INKEY$<>"" THEN GO TO 14
40
1450 PRINT AT 14,h+7;"
 ";AT 15,h+9;"                 "
1460 IF c$="n" OR c$="N" THEN GO
 TO 1610
1470 LET pc=pc+1
1480 GO SUB 570
1490 LET p=p+VAL a$(c,1 TO 2)
1500 IF VAL a$(c,1 TO 2)=11 THEN
 LET pa=pa+1
1510 LET c=c+1
1520 IF p<22 OR pa<=0 THEN GO TO
 1560
1530 LET pa=pa-1
1540 LET p=p-10
1559 GO TO 1520
1560 IF p>21 THEN GO TO 1580
1570 GO TO 1390
1580 PRINT AT 17,20; PAPER 7; FL
ASH 1;"REBENTOU"
1590 LET p=0
1600 GO TO 1640
1610 PRINT AT 15,20; PAPER 7;"So
ma "; FLASH 1;p
1620 IF pc>=5 THEN PRINT AT 15,h
+7; PAPER 7; FLASH 1;"[illegible]
[illegible]
1640 PAPER 7: FOR i=1 TO 9: PRIN
T AT i,1;"|        ": NEXT i
1650 PRINT AT 3,2; INK VAL i$(5)
;i$(3);AT 2,2;i$(4)
1660 IF p=0 THEN GO TO 1850
1670 IF p=21 AND pc=2 THEN GO TO
 1850
1680 LET v=1: LET h=3
1690 LET h=h+2
1700 IF d<22 OR da<=0 THEN GO TO
 1740
1710 LET da=da-1
1720 LET d=d-10
1730 GO TO 1700
1740 IF dc>=5 AND d>11 THEN GO T
O 1850
1750 IF d>=17 THEN GO TO 1850
1760 PAUSE 50
1770 PAUSE 1+50*(dc=2)
1780 PAPER 7
1790 GO SUB 570
1800 LET d=d+VAL a$(c,1 TO 2)
1810 IF VAL a$(c,1 TO 2)=11 THEN
 LET da=da+1
1820 LET c=c+1
1830 LET dc=dc+1
1840 GO TO 1690
1850 IF d>21 THEN LET d=0
1860 PAPER 7: INK 0
1870 LET h=dc*2
1880 PRINT AT 5,20;"SPEC ";
1890 PRINT AT 5,h+(22-(d=0));
1900 IF d=0 THEN PRINT AT 3,20;
FLASH 1;"REBENTEI"
1910 IF d>0 THEN PRINT ; FLASH 1
;d
1920 IF d=0 THEN LET dc=3
1930 IF p=0 THEN LET pc=3
1940 IF d=21 AND dc=2 THEN LET p
=0
1950 IF pc=2 AND p=21 THEN LET d
=0
1960 IF dc>=5 AND d>0 AND pc<5 T
HEN LET p=0
1970 IF pc>=5 AND p>0 AND dc<5 T
HEN LET d=0
1980 IF pc>=5 AND dc>=5 AND d>=p
 THEN LET p=0
1990 IF d>=p THEN PRINT AT 1,20;
 FLASH 1;"Ganhei": BEEP 1,-10: B
EEP .3,-20
2000 IF p>d THEN PRINT AT 15,20;
 FLASH 1;"Ganhou": FOR f=0 TO 30
: BEEP .01,f: BEEP .01,f+10: NEX
T f
2010 IF d>=p THEN LET st=st-bet
2020 IF d<p THEN LET st=st+bet
2030 IF pc>=5 AND p>0 THEN LET s
t=st+(bet/2)
2040 IF pc=2 AND p=21 THEN LET s
t=st+(bet/2)
2050 LET st=INT st
2060 PAUSE 100
2070 PAPER 4
2080 CLS
2090 IF st>0 THEN GO TO 930
2100 PAPER 7
2110 PRINT AT 6,3;"Perdeu todo o
 seu dinheiro"
2120 FOR i=1 TO 50: NEXT i
2130 PRINT AT 11,1;"qualquer tec
la para novo jogo";AT 14,3;"Use
""ENTER"" para terminar"
2140 LET v=CODE INKEY$
2150 IF v=0 THEN GO TO 2140
2160 PAPER 4
2170 CLS
2180 IF v=13 THEN GO TO 2210
2190 LET st=1000: LET x=x+1
2200 GO TO 930
2210 BORDER 1: PAPER 1: INK 7
2220 CLS
2230 PRINT AT 8,1;"ADEUS, foi um
 prazer ficar com";AT 10,1;"to[illegible]
 o seu dinheiro"
2240 PRINT AT 12,6;"Perdeu os ";
x*1000;"$"
2250 BEEP .1,(RND*10): GO TO 225
0
2260 RESTORE
2270 FOR f=USR "a" TO USR "i"+7
2280 READ a
2290 POKE f,a
2300 NEXT f
2310 DATA 0,0,0,0,0,0,0,0
2320 DATA 170,85,170,85,170,85,1
70,85
2330 DATA 0,56,56,16,214,254,214
,16
2340 DATA 0,16,56,124,254,124,56
,16
2350 DATA 170,170,85,85,170,170,
85,85
2360 DATA 0,108,254,254,124,124,
56,16
2370 DATA 0,16,56,124,254,254,21
4,16
2380 DATA 152,164,164,164,164,16
4,164,152
2390 DATA 128,128,128,128,128,12
8,128,128
2400 RETURN
```

OS GRAFICOS

Obtera os graficos quando o programa estiver todo passado, ou entao comece pela linha (2260) - logo que os queira utilizar faca "RUN" e volte a listagem.

♦- graphics "d"

♥- graphics "f"

♠- graphics "g"

♣- graphics "c"

10- graphics "h"

▒- graphics "b,e"

========
NEWBRAIN
========

****
G E N I N
****

in.PERSONAL COMPUTER,
N.7, Julho/1983
Trad. e adapt. por
ISABEL CRISTINA/Porto

SUBROTINA PARA ENTRADA E VALIDACAO DE DADOS
****

PARA O MICROCOMPUTADOR NEWBRAIN
****

NOTA: O numero de caracteres aceite em cada INPUT esta definido na linha 2020, na variavel "L", e a validacao do tipo de caracteres esta definida na variavel "IN$".

GENIN - e uma abreviatura para a rotina "GENERAL INPUT" e destina-se a ser usado como subrotina em programas onde sao necessarias grandes quantidades de INPUTs.
A rotina foi projectada para ser completamente a prova de erro, rejeitando INPUTs nao validos, enquanto admite o uso de "INSERT", "DELETE" e movimentos com o cursor para a esquerda e para a direita. A rotina verifica se o comprimento da string e menor ou igual ao valor de L e se cada caracter na string esta dentro da serie ASCII, definida por I1$. Estes valores sao para ser definidos previamente pelo programador. Exemplo:
Para aceitar palavras com 4 caracteres L deve ser igual a 4 e I1$ a "az".
Nesta listagem, nas linhas 2000 e 2020, encontra-se uma amostra para chamar a rotina. A presenca de "RETURN" faz regressar a string sem espacos e "CONTROL/CR" leva a string com espacos para a string de comprimento L. O INPUT volta a IN$.
Se usar um INPUT numerico, entao junte a linha 925 IN=VAL(IN$).
Esperamos que esta subrotina venha a ser util na execucao dos seus programas e aproveitamos para por a questao

ONDE E QUE ESTAO OS PROGRAMADORES DO NEWBRAIN?

Esperamos os vossos programas!

=========
G E N I N
=========

```
1 REM PROGRAMA .............."NEW BRAIN"
10 GOTO 2000
500 REM ** GENIN **
510 IF LEN(I1$)<>2 THEN PRINT"no I1$":RETURN
520 OPEN#5,5:PUT 21:GETx1,y1:f1=0:f2=0:IN$=""
530 L1=0:PUT6:LO=ASC(LEFT$(I1$,1)):HI=ASC(RIGHT$(I1$,1))
532 IF LO>HI THEN RETURN
540 GET#5,9$:IF 9$=""THEN 540
550 9=ASC(9$)
560 IF 9=13 THEN 900
570 IF 9=3 THEN f2=1:GOTO 900
580 IF 9=8 THEN f1=0:GOTO 690
590 IF 9=26 THEN f1=0:GOTO 670
600 IF 9=17 THEN f1=1:GOTO 540
610 IF 9=24 THEN 800
620 IF 9<32 THEN 540
630 IF 9<LO OR 9>HI THEN 540
635 IF f1=1 THEN GOSUB 730:GOTO 540
640 PRINT 9$;:L1=L1+1: IFL1=L THEN L1=L1-1:PUT 8
650 GOTO 540
660 REM ** cursor direito **
670 IF L1=L-1 THEN 540
680 GOTO 640
690 REM ** cursor esquerdo **
700 IF L1=0 THEN 540
710 L1=L1-1:PUT 8:GOTO 540
720 REM ** insert **
730 PUT 7:FOR s=x1+L -2 TO x1+L1STEP-1:PUT22,s,y1,20:GET ch
740 PUT 22,s+1,y1,ch:NEXT s:PUT22,s+1,y1,9:PUT8,6
750 RETURN
800 PUT 7:IF L1=0 THEN 850
810 fi=x1+L-1:FOR s=x1+L1 TO fi:PUT 22,s,y1,20:GET ch
820 PUT 22,s-1,y1,ch: NEXT s:L1=L1-1
830 PUT 22,fi,y1,32:PUT22,x1+L1,y1,6:GOTO 540
850 FOR s=x1 TOx1+L-2:PUT22,s+1,y1,20:GET ch
860 PUT22,s,y1,ch:NEXT s:PUT22,x1+L-1,y1,32
870 PUT22,x1,y1,6:GOTO 540
900 PUT7:FOR s=0 TO L-1:PUT22,x1+s,y1,20:GET ch:IN$=IN$+CHR$(ch):NEXT s
910 IF f2=1 THEN 930
920 IF RIGHT$(IN$,1)=CHR$(32) THENL=L-1:IN$=LEFT$(IN$,L):GOTO 920
930 CLOSE#5:RETURN
2000 PUT31:PRINT "nome [        ]"
2020 L=8:I1$=" z":PUT22,7,0:GOSUB 500:PRINT:PRINTIN$:END
```

**COMPUTADOR NEWBRAIN

Adaptação - Isabel Cristina

RELÓGIO

Se introduzir este Programa , terá o seu NewBrain a funcionar como um relógio.

Depois de introduzir a hora ,os minutos e os segundos(adiantar 2 para coordenar com o compasso de espera) faça RUN e para acabar STOP/NEWLINE .

```
5 PUT 31
10 PUT22,5,5:INPUT("QUE HORAS SAO ? ")E
20 F=INT(E/10):E=E-(F*10)
30 PUT22,5,7:INPUT("QUANTOS MINUTOS ?  ")C
40 D=INT(C/10):C=C-(D*10)
50 PUT22,5,9:INPUT("QUANTOS SEGUNDOS  ?  ")A
60 B=INT(A/10):A=A-(B*10)
70 FOR N=1 TO 300:NEXT N
80 A=A+1
90 IF A>9 GOTO 110
100 GOTO 300
110 A=0
120 B=B+1
130 IF B>5 GOTO 150
140 GOTO 300
150 B=0
160 C=C+1
170 IF C>9 GOTO 190
180 GOTO300
190 C=0
200 D=D+1
210 IF D>5 GOTO 230
220 GOTO 300
230 D=0
240 E=E+1
250 IF E>9 GOTO 270
260 GOTO 290
270 E=0
280 F=F+1
290 IF A=0 THEN 300
292 IF B=0 THEN 300
294 IF C=0 THEN 300
296 IF D=0 THEN 300
297 IF E=1 THEN 300
298 IF F=0 THEN 300
300 PUT 31
310 PUT22,10,12:PRINT F;E;":";D;C;":";B;A
315 PUT22,9,6:?"** RELOGIO NEW BRAIN **"
320 GOTO 70
```

P R O G R A M A C A O   E S T R U T U R A D A
(Continuacao)

in. "STRUCTURED COBOL", PHILIPPAKIS Andreas e KAZMIER Leonard

Traduzido e Adaptado ao BASIC por
FRANCISCO J.R. SANTOS

Podemos entender um programa grande abstraindo-nos dos detalhes de cada modulo, no seu objectivo e funcoes, debrucando-nos apenas nas relacoes entre os modulos. Na pratica, desenvolvemos uma estrutura modular, procedendo do geral para o especifico, ou usando o que e largamente conhecido como "top-down aproach".

Note-se que no processo de desenvolvimento da estrutura do programa nao usamos linguagem de programacao de computadores. Escrevemos os nossos pensamentos numa linguagem comum, com ou sem graficos auxiliares, tais como setas, quadrados, parentesis, etc.

Um termo largamente usado e a pseudocodificacao, que vigora para escrever programas em linguagem nao programatica.

Alguns praticantes da pseudocodificacao adiantaram-se ao mencionar regras de escrita da pseudocodificacao. O nosso senso comum guia-nos outra vez, dizendo-nos como usar qualquer linguagem comum que achemos util. Para exemplificar, numa grande instalacao podem ser beneficas formas standard de pseudocodificacao.

A pseudocodificacao pode ser escrita em varios graus de abstraccao; em geral aconselhamos a descricao global de todo o trabalho numa pagina, e entao a descricao de cada modulo que deve ser feita em paginas separadas. Esta descricao pode servir como documentacao parcial, ou mesmo completa, de um programa.

Apresentamos ate aqui as caracteristicas ou principios gerais da programacao estruturada. Sintetizando:

- um programa deve ser compreensivel
- um programa deve ser correcto
- um programa deve incorporar uma simples e direita linha de fluxo logico
- um programa deve compor-se de modulos interligados, sendo cada modulo simples e claro, como deve ser concebido para o trabalho.

APENDICE A

Em relacao ao que foi descrito no capitulo ESTRUTURA DO PROGRAMA, posso ainda acrescentar que estas ideias sao muito usadas nos meus programas, relativamente as quais me tentei aperfeicoar. Nao digo isto para que deem credibilidade ao meu trabalho,

mas apenas para provar que estou a vontade para o que em seguida vou explicar.

Nos meus programas uso sempre pelo menos dois tipos de rotinas:

1) Uma rotina principal, que chama as rotinas secundarias

Ex:

```
10 REM "EXPERIENCIA"
20 GO SUB 1000: REM ECRAN
30 GO SUB 2000: REM MOV.PRINCIPAL
40 GO SUB 3000: REM FINAL
```

Esta rotina pode ser por exemplo aplicada a um ZX SPECTRUM. Mas no ZX81, que nao funciona com "multistatements", as REMs podem ser desdobradas para outras linhas:

```
10 REM "EXPERIENCIA"
15 REM ECRAN
20 GOSUB 1000
25 REM MOV.PRINCIPAL
30 GOSUB 2000
35 REM FINAL
40 GOSUB 3000
```

2) A seguir escrevem-se as rotinas secundarias, ainda que estas possam chamar outras, mesmo dependentes daquelas.

Ex:

```
1000 REM ****ECRAN****
1010 GO SUB 5000
1999 RETURN
```

E assim fariamos para as outras sub-rotinas ate chegar a 5000 que, como e obvio, e ja um modulo mais pequeno.

O sistema e dividir um programa em subrotinas pequenas, de modo a que cada uma tenha uma funcao bem definida e clara.

Depois de escrevermos e testarmos cada uma das sub-rotinas, podemos finalmente interliga-las com GOSUBs, tal como esta feito no primeiro ponto deste apendice.

O nome "EXPERIENCIA" e apenas o nome do programa, e constitui a rotina principal, que nada tem a ver com o MOVIMENTO PRINCIPAL. E onde esta o ECRAN pode estar outro nome tal como INICIO.

FORMAS DA ESTRUTURA DO PROGRAMA

Um programa estruturado pode ser completamente desenvolvido usando 3 formas basicas da estrutura do programa que sao:

- Sequencia
- If-then-else
- Do-while

A estrutura SEQUENCIA indica um fluxo de programa logico. Cada bloco pode consistir numa frase (statement); pode consistir num modulo por inteiro, ou mesmo uma coleccao de modulos.

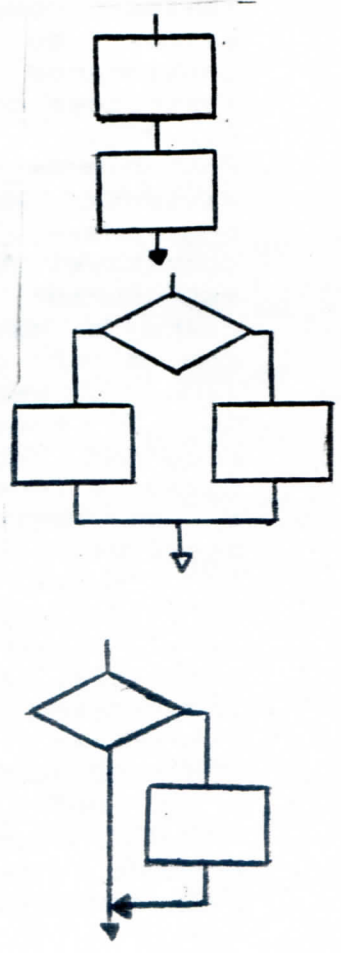
Em BASIC ilustramos com frases (statements) em serie tais como:

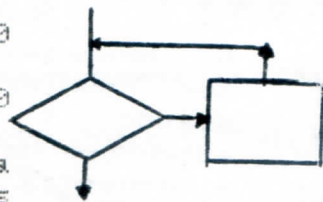
```
LET TOTAL=SUBTOTAL+TOTAL
PRINT TOTAL
```

A estrutura IF-THEN-ELSE indica um fluxo condicional do programa. Assim, o programa toma um ou outro caminho, dependendo a condicao geralmente referida como um predicado - e falso ou verdadeiro.

No BASIC ilustramos com a estrutura seguinte:

```
IF AMOUNT 100 THEN LET PRICE=PRICE*0.20
ELSE LET PRICE=PRICE*0.10
```

Podemos ter casos tais que se a condicao e tomada, faz-se algo, se nao continuamos com a sequencia do programa. Em BASIC pode-se executar a

estrutura com a omissao do ELSE como se segue:

```
IF AMOUNT 100
LET
PRICE=PRICE*0.20
```

A estrutura DO-WHILE da-nos a possibilidade de uma operacao em ciclo (loop), que e uma execucao repetitiva de um segmento de um programa. Um ciclo nao e infinito, se for correcto. O predicado e testado por uma condicao. Se a condicao for tomada como valida, entao saimos do ciclo; se nao, continuamos a executar as instrucoes contidas no ciclo.
(...)

Posteriormente, nos queremos salientar que GO TO e menos programavel, e nao e muito compativel com a programacao estruturada, uma estrutura clara e que pode ser cumprida com o uso do GO TO. Por outro lado, o uso indiscriminado do GO TO pode-nos levar a um programa labirinto, e por esta razao e preferivel restringir o uso deste verbo sempre que possivel.

As tres estruturas atras referidas serao suficientes para representar muitos programas. E ainda conveniente apresentar mais duas estruturas: DO UNTIL e CASE (ON X GO TO A, B, C).

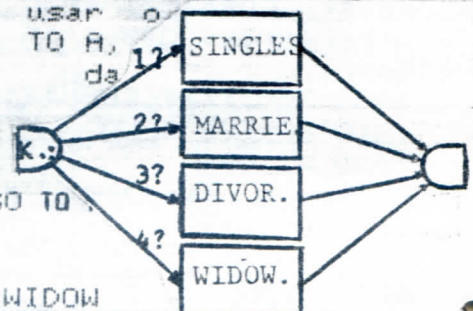
O DO UNTIL assemelha-se ao DO WHILE, excepto que o bloco precede o teste. Assim o bloco sera executado pelo menos uma vez.

NOTA:
Como no prefacio referi, originalmente este artigo foi escrito para a linguagem COBOL. Na adaptacao nao quis retirar as instrucoes DO WHILE e DO UNTIL, visto o BASIC de alguns computadores (tal como o LYNX) ter estas instrucoes.

Vamos admitir que SINGLES, MARRIED, DIVORCED e WIDOW eram variaveis que anteriormente ja estavam iniciadas. Onde esta ADD 1 TO (variavel), devera ser tida em BASIC na seguinte forma:
LET (variavel)=(variavel)+1
uma vez que as instrucoes estao inscritas em COBOL.

A estrutura CASE (ON X GO TO A, B, C) e conveniente quando temos um numero grande de alternativas a serem testadas. Supondo que o estado material de cada empregado e codificado em 1, 2, 3 e 4 correspondendo respectivamente a solteiro, casado, divorciado e viuvo (deve ser pre-verificada a correctidao dos codigos). Desejamos contar o numero de empregados pertencentes a cada classe. Assim temos:

Assumindo que KODE contem o estado de cada empregado, podemos usar o ON X GO TO A, B, C da seguinte maneira

ON X GO TO SINGLES, MARRIED, DIVORCED, WIDOW

```
SINGLES.  ADD 1 TO SINGLES-COUNT
          GO TO NEXT-P.
MARRIED.  ADD 1 TO MARRIED-COUNT
          GO TO NEXT-P.
DIVORCED.ADD 1 TO DIVORCED-COUNT
          GO TO NEXT-P.
WIDOWED.  ADD 1 TO WIDOWED-COUNT
          GO TO NEXT-P.
```

Neste exemplo, a estrutura aparece grosseira. A razao e que cada um dos quatro paragrafos fazem coisas simples. Se imaginassemos um trabalho mais longo a executar, para cada caso, contudo poderiamos ter entao uma estrutura muito mais atractiva.

(Continua no proximo numero)

****

******SOFTWARE******

SINCLAIR SPECTRUM

OS SÓCIOS DO CLUBE Z-8Ø BENEFICIARÃO
DE UM DESCONTO DE 20 %
NA COMPRA DE SOFTWARE DISPONÍVEL.

Contas Correntes (48K RAM)..............................1 000.00

Analise Investimentos (48K RAM).........................1 000.00

Stocks (48K RAM) (500 Produtos).........................1 200.00

VU-CALC (16K ou 48K RAM) Folha de calculo...............1 000.00

VU-FILE (16K ou 48K RAM) Ficheiros......................1 000.00

VU-3D (48K RAM) Desenho a tres dimensoes................1 000.00

PERT/CPM (48K RAM)...................................... 800.00

Matematica (48K RAM).................................... 800.00

Estatistica (48K RAM)................................... 800.00

Biblioteca (48K RAM).................................... 600.00

Indice Bibliografico (48K RAM).......................... 600.00

SPEc-8 (Inventario Permanente-250 itens) (48K RAM).......1 000.00

Compiler (48K RAM)......................................1 000.00

Assembler/Disassembler (16K ou 48K RAM)................. 600.00

Calculo de Porticos (48K RAM)...........................2 000.00

SINCLAIR SPECTRUM (Cont)

JOGOS-I (Invaders,Laser,Lunar Lander,Alien,Mastermind,Maze)
(16K RAM)..............................1 000.00
JOGOS-II (Batalha Naval,Space Roller,Ski,Comando,Holocausto,
Damas) (16K RAM)..............................1 000.00
Sun-Puzzle (16K RAM)........................ 600.00
Travessia (16K RAM)........................ 600.00
Blind Alley (16K RAM)........................ 600.00
Intruders (16K RAM)........................ 600.00
Cyber Rates (16K RAM)........................ 600.00
SPEc-1 (Cercado,Planeta Grool) (16K RAM)............ 600.00
SPEc-2 (Campo de Minas,Aterragem) (16K RAM)........... 600.00
SPEc-3 (Kingkong,Salvamento) (16K RAM)............ 600.00
SPEc-5 (Vingador) (16K RAM)........................ 600.00
SPEc-6 (Fuga) (16K RAM)........................ 600.00
Horace (16K RAM)........................ 600.00
Sim.Voo Nocturno (16K RAM)........................ 600.00
Planetoids (16K ou 48K RAM)........................ 600.00
Ground Attack (16K ou 48K RAM)........................ 600.00
Arcadia (16K RAM)........................ 600.00
Slippery Sid (16K RAM)........................ 600.00
The Hobbit-Jogo de Aventuras (48K RAM)..................1 000.00
Simulador de Voo (48K RAM)........................ 600.00
Xadrez (48K RAM)........................ 600.00
QS Xadrez (48K RAM)........................ 600.00
Time Gate (48K RAM)........................ 600.00
Orbiter (48K RAM)........................ 600.00
Meteor Storm (48K RAM)........................ 600.00
Horizons (48K RAM)........................ 600.00
Tunel (48K RAM)........................ 600.00
SPEc-4 (E.T.) (48K RAM)........................ 600.00
Penetrator (48K RAM)........................ 600.00
Mined Out (48K RAM)........................ 600.00
Trader (48K RAM)........................ 600.00
Batalha (48K RAM)........................ 600.00
Monopoly (48K RAM)........................ 600.00
Castelo (48K RAM)........................ 600.00
Futebol (48K RAM)........................ 600.00
Derbi Day (48K RAM)........................ 600.00
The Orb (48K RAM)........................ 600.00
Crusing (48K RAM)........................ 600.00

LIVROS

A lista de livros existentes no CLUBE Z-8Ç está publicada nos boletins nºs. 6 (pág.4), 7 (pág.6) e 8 (pág.8).